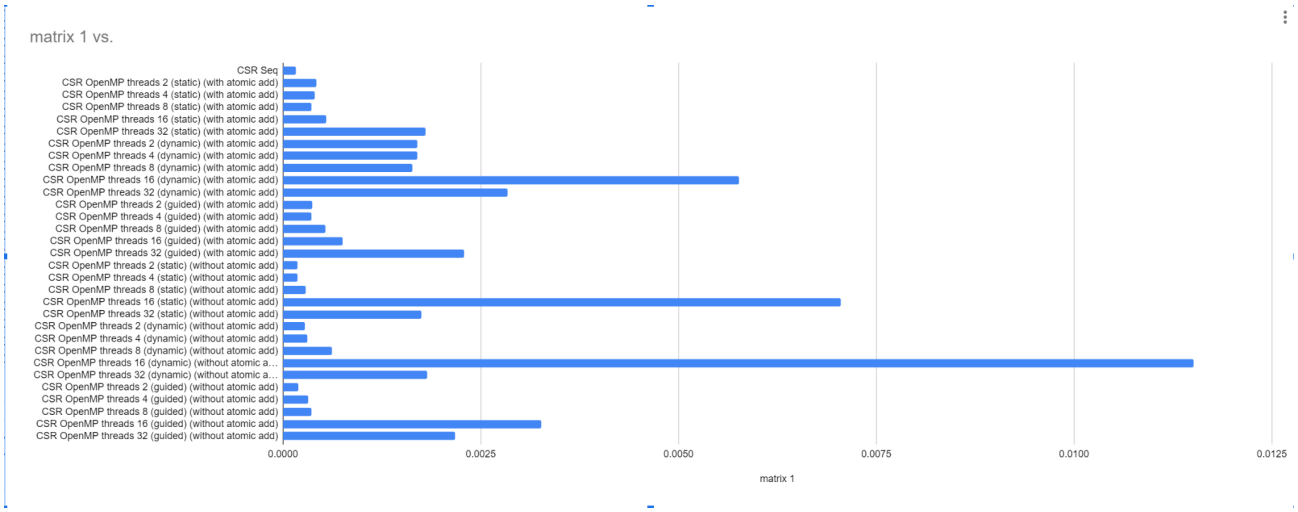
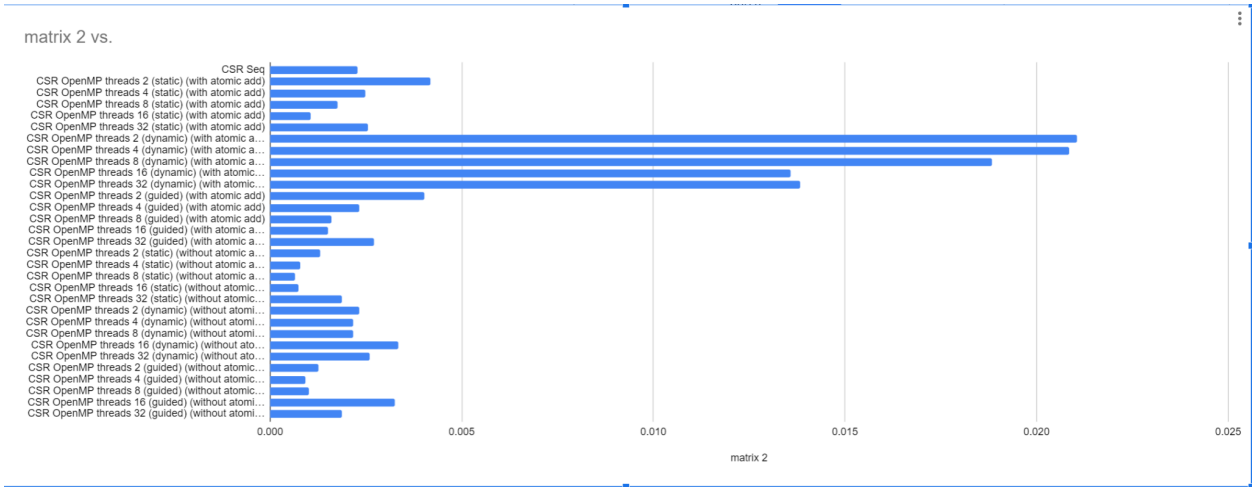
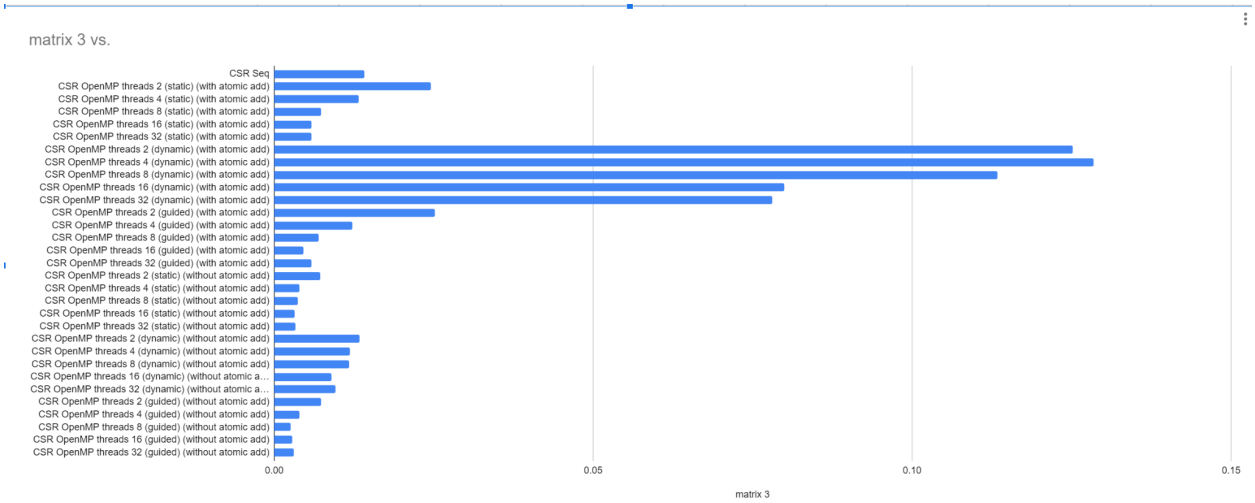
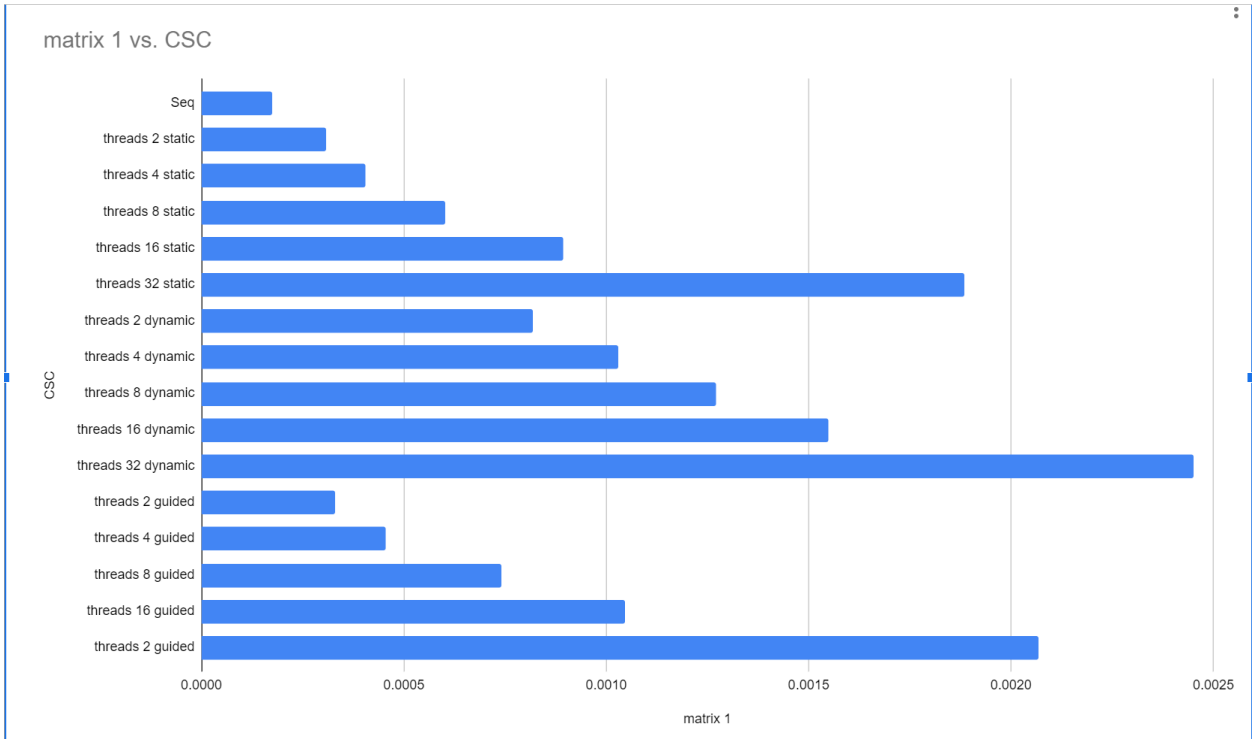
1. To run the code, just type ./run.sh in terminal under the folder
2. Execution time(in later page)
3. Testing Platform: intel i7-1170
4. Analysis for CSRseq and CSROpenMP (static dynamic guided)
   The result chart is in pictures below:
   We can see that if we implement atomic add, the result will be improved in matrix 1 and in matrix 2 and 3, without atomic add, the performance will be improved. I believed that with increasing of the size of the matrix, doing inner product will need to have atomic operation to ensure performance, otherwise the compiler control atomic operation will cost more time. In matrix 3, guided schedule policy works best, in matrix 1, static schedule works best. In all the experiments, the dynamic without atomic add will perform worst performance.

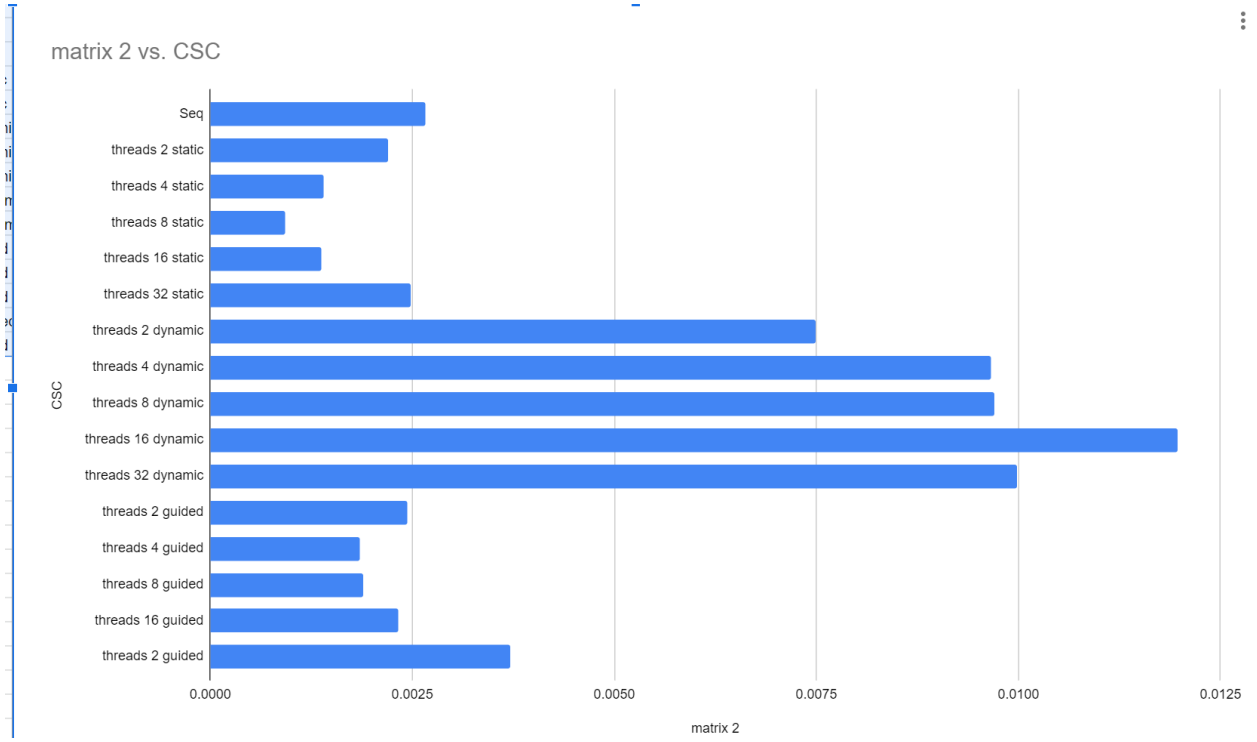


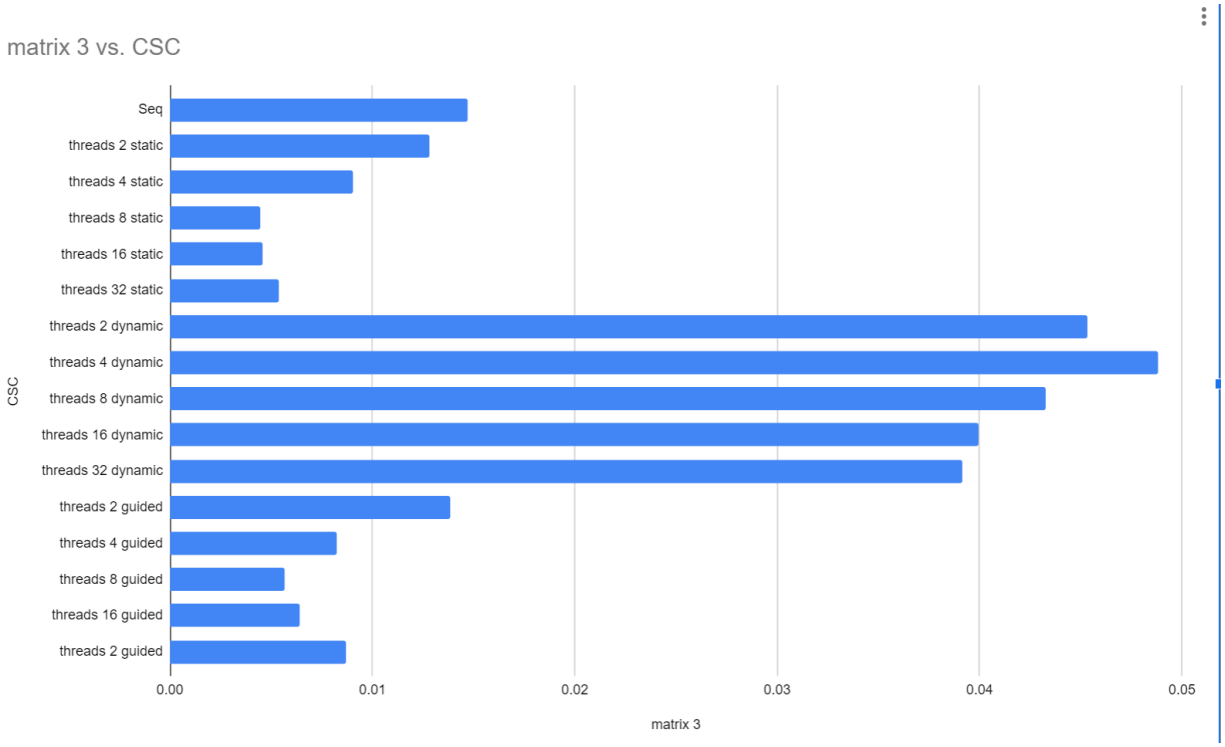




5. Analysis for CSRseq and CSROpenMP (static dynamic guided)
   I am using outer product in CSC format (without and reduction in algorithm, so I use atomic add all the time), from the picture below, we can see that in matrix 1, we can't gain any speedup with parallelism, I think that is because matrix size is not big enough, so the atomic operation costs more in parallel programing.
   However, when the size increase, we can see clearly the speed up in outer product, and static schedule always perform better among three of them.







6. Analysis for CSC and CSR
   From the result we can see that CSR with inner product performance is much worse than CSC with outer product, I believe that is for the large matrix with highly sparsity, outer product can reduce cache replacement and improve the performance greatly, that is the reason why people prefer outer product in highly sparse DNN accelerator design.

Result Table:

	matrix 1	matrix 2	matrix 3
CSR Seq	0.000164	0.002291	0.014098
CSR OpenMP threads 2 (static) (with atomic add)	0.00042	0.004192	0.024568
CSR OpenMP threads 4 (static) (with atomic add)	0.0004	0.002476	0.013261
CSR OpenMP threads 8 (static) (with atomic add)	0.000357	0.001749	0.007415
CSR OpenMP threads 16 (static) (with atomic add)	0.000541	0.001065	0.005907
CSR OpenMP threads 32 (static) (with atomic add)	0.0018	0.002551	0.00591
CSR OpenMP threads 2 (dynamic) (with atomic add)	0.001701	0.02106	0.125188
CSR OpenMP threads 4 (dynamic) (with atomic add)	0.001696	0.020859	0.128485
CSR OpenMP threads 8 (dynamic) (with atomic add)	0.00164	0.018826	0.113321
CSR OpenMP threads 16 (dynamic) (with atomic add	0.005766	0.013586	0.079922
CSR OpenMP threads 32 (dynamic) (with atomic add	0.002838	0.013828	0.078104
CSR OpenMP threads 2 (guided) (with atomic add)	0.000367	0.00402	0.025172
CSR OpenMP threads 4 (guided) (with atomic add)	0.000359	0.002326	0.012256
CSR OpenMP threads 8 (guided) (with atomic add)	0.000538	0.001608	0.006987
CSR OpenMP threads 16 (guided) (with atomic add)	0.000757	0.001503	0.004548
CSR OpenMP threads 32 (guided) (with atomic add)	0.002291	0.002722	0.005852
CSR OpenMP threads 2 (static) (without atomic add)	0.000183	0.001309	0.007214
CSR OpenMP threads 4 (static) (without atomic add)	0.000179	0.000777	0.003958
CSR OpenMP threads 8 (static) (without atomic add)	0.00029	0.000651	0.003715
CSR OpenMP threads 16 (static) (without atomic add	0.007052	0.000747	0.003194
CSR OpenMP threads 32 (static) (without atomic add	0.001751	0.00187	0.003346
CSR OpenMP threads 2 (dynamic) (without atomic ac	0.000276	0.002319	0.013428
CSR OpenMP threads 4 (dynamic) (without atomic ac	0.000309	0.002168	0.011836
CSR OpenMP threads 8 (dynamic) (without atomic ac	0.000617	0.002179	0.011718
CSR OpenMP threads 16 (dynamic) (without atomic a	0.011513	0.003348	0.009028
CSR OpenMP threads 32 (dynamic) (without atomic a	0.001817	0.002589	0.009616
CSR OpenMP threads 2 (guided) (without atomic add	0.00019	0.001252	0.007302
CSR OpenMP threads 4 (guided) (without atomic add	0.000319	0.00093	0.003909
CSR OpenMP threads 8 (guided) (without atomic add	0.000359	0.00102	0.002604
CSR OpenMP threads 16 (guided) (without atomic ad	0.003268	0.003246	0.002834
CSR OpenMP threads 32 (guided) (without atomic ad	0.002175	0.001865	0.003119

A	B	C	D
CSC	matrix 1	matrix 2	matrix 3
Seq	0.000174	0.002656	0.014731
threads 2 static	0.000308	0.002199	0.012803
threads 4 static	0.000405	0.0014	0.009031
threads 8 static	0.000602	0.000927	0.004447
threads 16 static	0.000893	0.001369	0.004585
threads 32 static	0.001884	0.002475	0.005386
threads 2 dynamic	0.00082	0.007496	0.045304
threads 4 dynamic	0.001031	0.009654	0.048837
threads 8 dynamic	0.001271	0.009704	0.043242
threads 16 dynamic	0.001549	0.011963	0.039958
threads 32 dynamic	0.002451	0.009988	0.039151
threads 2 guided	0.000331	0.00244	0.013862
threads 4 guided	0.000455	0.001853	0.008221
threads 8 guided	0.00074	0.001892	0.005654
threads 16 guided	0.001046	0.00233	0.006397
threads 2 guided	0.002069	0.003717	0.008696